

### Listing of Claims

1 Claim 1 (Previously Presented): A method of implementing atomic transactions in a  
2 system, said method comprising:

3 requesting in a user program a transaction identifier for an atomic transaction;  
4 generating said transaction identifier in a transaction manager in response to said  
5 requesting;

6 specifying in said user program a plurality of combinations, wherein each of said  
7 plurality of combinations contains said transaction identifier, a task procedure, and a rollback  
8 procedure, wherein said task procedure implements a part of said atomic transaction and said  
9 rollback procedure is designed to rollback said task procedure;

10 executing a set of task procedures in a sequential order according to said user program,  
11 wherein said set of task procedures are contained in said task procedures specified in said  
12 plurality of combinations;

13 keeping track of a set of rollback procedures corresponding to said set of task  
14 procedures, each of said set of procedures being determined based on a combination  
15 corresponding to an executed task procedure contained in said set of task procedures, said  
16 combination being contained in said plurality of combinations specified in said user program;  
17 and

18 executing said set of rollback procedures in a reverse order of said sequential order if  
19 said atomic transaction is to be aborted,

20 wherein said rollback procedure is specified as a separate procedure from said task  
21 procedure in said user program.

1 Claim 2 (Original): The method of claim 1, wherein said transaction identifier is  
2 unique to each of the atomic transactions.

1 Claim 3 (Previously Presented): The method of claim 1, wherein said keeping  
2 comprises storing data representing said rollback procedures in a stack.

1 Claim 4 (Original): The method of claim 3, wherein said stack is stored in a memory.

1 Claim 5 (Original): The method of claim 1, further comprising examining a status

2 returned by execution of one of said task procedures and performing said aborting if said  
3 status indicates an error.

1 Claim 6 (Original): The method of claim 1, wherein said aborting is performed  
2 asynchronously.

1 Claims 7 (Previously Presented): A computer readable medium carrying one or more  
2 sequences of instructions representing a user program for execution on a system, said user  
3 program implementing an atomic transaction, wherein execution of said one or more  
4 sequences of instructions by one or more processors contained in said system causes said one  
5 or more processors to perform the actions of:

6 requesting an identifier in said user program from a transaction manager for said  
7 atomic transaction, wherein said transaction manager generates a unique value as said  
8 identifier;

9 setting a variable to equal said identifier;

10 specifying a plurality of combinations in said user program for execution in said  
11 system, wherein each of said plurality of combinations contains said variable, a task  
12 procedure, and a rollback procedure, wherein said task procedure implements a part of said  
13 atomic transaction and said rollback procedure is designed to rollback said task procedure,  
14 wherein said variable in each of said plurality of combinations specifies said identifier  
15 generated by said transaction manager; and

16 aborting said atomic transaction by specifying said identifier associated with an abort  
17 procedure to cause said rollback procedures to be executed,

18 wherein said plurality of combinations and said abort procedure are contained in said  
19 user program.

1 Claim 8 (Original): The computer readable medium of claim 7, wherein said specifying  
2 comprises including each of said plurality of combinations in a single procedure call.

1 Claim 9 (Original): The computer readable medium of claim 7, further comprising  
2 examining a status returned by execution of one of said task procedures and performing said  
3 aborting if said status indicates an error.

1 Claim 10 (Previously Presented): A computer readable medium carrying one or more  
2 sequences of instructions for supporting implementation of an atomic transaction in a system,  
3 wherein execution of said one or more sequences of instructions by one or more processors  
4 contained in said system causes said one or more processors to perform the actions of:

5 generating an identifier for said atomic transaction for a user program;

6 receiving a plurality of combinations for execution from said user program, wherein  
7 each of said plurality of combinations contains said transaction identifier, a task procedure,  
8 and a rollback procedure, wherein said task procedure implements a part of said atomic  
9 transaction and said rollback procedure is designed to rollback said task procedure;

10 executing a set of task procedures in a sequential order according to said user program,  
11 wherein said set of task procedures are contained in said plurality of combinations;

12 keeping track of a set of rollback procedures corresponding to said set of task  
13 procedures, each of said set of procedures being determined based on a combination  
14 corresponding to an executed task procedure contained in said set of task procedures, said  
15 combination being contained in said plurality of combinations specified in said user program;  
16 and

17 executing said set of rollback procedures in a reverse order of said sequential order in  
18 response to receiving an abort request,

19 wherein said rollback procedure is specified as a separate procedure from said task  
20 procedure in said user program.

1 Claims 11 - 12 (Canceled)

1 Claim 13 (Previously Presented): The computer readable medium of claim 10, wherein  
2 said transaction identifier is generated to be unique for each atomic transaction.

1 Claim 14 (Previously Presented): The computer readable medium of claim 10, wherein  
2 said set of rollback procedures are represented in the form of a stack.

3 Claim 15 (Original): The computer readable medium of claim 14, wherein said stack  
4 is stored in a memory.

1 Claim 16 (Previously Presented): A computer system comprising:  
2 a memory storing a plurality of instructions; and  
3 a processing unit coupled to said memory and executing said plurality of instructions  
4 to support implementation of atomic transactions in a programming environment, said  
5 processing unit being operable to:

6 request in a user program transaction identifier for an atomic transaction;  
7 generate said transaction identifier in a transaction manager in response to said  
8 requesting, wherein said transaction manager is provided external to said user  
9 program;

10 specify in said user program a plurality of combinations wherein each of said  
11 plurality of combinations contains said transaction identifier, a task procedure, and a  
12 rollback procedure, wherein said task procedure implements a part of said atomic  
13 transaction and said rollback procedure is designed to rollback said task procedure,  
14 wherein said rollback procedure is specified as a separate procedure from said task  
15 procedure;

16 execute a set of task procedures in a sequential order according to said user  
17 program;

18 keep track of a set of rollback procedures corresponding to said set of task  
19 procedures, each of said set of procedures being determined based on a combination  
20 corresponding to an executed task procedure contained in said set of task procedures,  
21 said combination being contained in said plurality of combinations specified in said  
22 user program; and

23 execute said set of rollback procedures in a reverse order of said sequential  
24 order if said atomic transaction is to be aborted, wherein said rollback procedures are  
25 identified according to said keeping.

1 Claim 17 (Original): The computer system of claim 16, wherein said transaction  
2 identifier is unique to each of the atomic transactions.

1 Claim 18 (Previously Presented): The computer system of claim 16, wherein said  
2 processing unit is operable to store data representing said rollback procedures in a stack to

3 perform said keep.

1 Claim 19 (Original): The computer system of claim 18, wherein said stack is stored in  
2 a memory.

1 Claim 20 (Original): The computer system of claim 16, wherein said processing unit  
2 is further operable to examine a status returned by execution of one of said task procedures  
3 and to perform said aborting if said status indicates an error.

1 Claim 21 (Previously Presented): The computer system of claim 16, wherein said  
2 processing unit is operable to execute said rollback procedures asynchronously.

1 Claims 22 - 24 (Canceled)

1 Claim 25 (Previously Presented): The computer readable medium of claim 7, wherein  
2 said rollback procedure is specified as a separate procedure from said task procedure in said  
3 user program.